4

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/032,762	12/26/2001	Mark A. Schmisseur	42390P12808	5769

45459 7590 10/02/2006

GROSSMAN, TUCKER, PERREAULT & PFLEGER, PLLC
C/O PORTFOLIO IP
P. O. BOX 52050
MINNEAPOLIS, MN 55402

EXAMINER	
RAHMAN, FAHMIDA	

ART UNIT	PAPER NUMBER
2116	

DATE MAILED: 10/02/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	10/032,762	SCHMISSEUR ET AL.
	Examiner	Art Unit
	Fahmida Rahman	2116

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 August 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3,5-10,12-18,20-23,25-32,34 and 35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,5-10,12-18,20-23,25-32,34 and 35* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *01 April 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the arguments filed on 8/8/2006.

2. Claims 1, 9, 15, 22, 29 have been amended, claims 4, 11, 19, 24, 33 have been canceled, no new claims have been added. Thus, claims 1-3, 5-10, 12-18, 20-23, 25-32, 34-35 are pending.

Claim Rejections - 35 USC § 112

Claims 3, 5, 12, 25, 32 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 3 recites the limitation “a system memory” in lines 1-2 and 3. It is not appeared whether it is same or different from the recitation of “a system memory” in line 11 of claim 1. It is necessary to establish a relationship between the two recitations. For the rest of the action, it is assumed that same system memory was intended.

Claim 5 recites the limitation “an address translation unit” in lines 2-3. It is not appeared whether it is same or different from the recitation of “address translation unit” in line 12 of claim 1. It is necessary to establish a relationship between the two recitations. For the rest of the action, it is assumed that same address translation unit was intended.

Claim 12 recites the limitation “an address translation unit” in line 2. It is not appeared whether it is same or different from the recitation of “address translation unit” in line 8 of claim 9. It is necessary to establish a relationship between the two recitations. For the rest of the action, it is assumed that same address translation unit was intended.

Claim 25 recites the limitation “an address translation unit” in line 3. It is not appeared whether it is same or different from the recitation of “address translation unit” in line 9 of claim 22. It is necessary to establish a relationship between the two recitations. For the rest of the action, it is assumed that same address translation unit was intended.

Claim 32 recites the limitation “a boot address” in line 2. It is not appeared whether it is same or different from the recitation of “a boot address” in line 6 of claim 29. It is necessary to establish a relationship between the two recitations. For the rest of the action, it is assumed that same boot address was intended.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-3, 5-7, 9-10, 12-13, 15-18, 20, 22-23, 25-27, 29-32, 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Anderson (U.S. 5,898,869), in view of Futral et al (US Patent 5925099).

Regarding claim 1, **Anderson discloses a system comprising:**

- **a core processing circuit** (13 comprises core processor 31), **and**
- **a host processing system** (11) **coupled to the core processing circuit through a host bridge** (17), **the host processing system comprising**:
- **logic to maintain the core processing circuit in a reset state** (column 2, lines 65-66, column 3, lines 6-7) **during power up of the core processing system** (lines 4-6 of column 3 mention that the resetting comprises asserting power-on reset signal. In addition, lines 9-11 of column 3 mention that the releasing of the suspended processor from reset includes host accessing an address. Thus, the core processing system is maintained in a reset state during power up until released by the host) **and**
- **logic to load a reset vector to one or more registers at a boot address associated with the core processing circuit** (column 6, lines 13-15 in view of column 8, lines 7-9; Anderson discloses the reset vector as comprising an "initial program" which necessarily must be loaded to registers), **the reset vector comprising one or more instructions to fetch additional instructions** (column 8, lines 24-26. Anderson discloses that the boot sequence can include a loader program to download a program from host 11 to FLASH memory, which

necessarily must include fetching additional instructions) **to initialize the core processing** circuit (lines 14-16 of column 8 mention that the boot sequence completes the initialization process) **upon release from the reset state** (lines 5-15 of column 6 mention that the booting is performed upon releasing from reset state)

- **wherein the host processing system further comprises a system memory** (host can be a laptop computer according to line 15 of column 4. Thus, it has a system memory) **and logic to set an address translation unit** ("address decode circuitry" mentioned in line 48 of column 2 is set by host for causing the processor to boot. The booting occurs from the downloaded code within dual port memory as mentioned in lines 5-15 of column 6) **to enable at least one outbound transaction to address at least one location in the system memory to fetch instructions from the system memory in response to requests from the core processing unit** (the host sets the address decode circuitry with a particular address for releasing suspended processor execution and causing the processor to boot (lines 30-33 of column 9). The boot program may include diagnostic routine, which can cause further diagnostic routine to be loaded from host as mentioned in lines 28-35 of column 8. Thus, host sets address decode circuitry or address translation unit to activate booting, which again causes outbound transaction to address the memory to fetch instructions from host memory in response to request from the core processing circuit).

Although Anderson fetches additional code from memory, Anderson does not explicitly disclose an address translation unit addressed to the system memory, however said transactions would necessarily have to be present in order for the host processing system to properly communicate with the core processing circuit.

Futral et al teach a system where ATU (218) is configured to convert an internal data bus address (204) associated with the outbound transaction to an external data bus address (208) and configured to forward the outbound transaction from an internal data bus (lines 47-67 of column 4) coupled to the core processing circuit (202) to an external data bus coupled to the system memory (201).

It would have been obvious for one ordinary skill in the art to incorporate the ATU within Anderson as proper communication between PCMCIA and host requires translating core bus transaction to host bus transaction.

Regarding claim 2, Anderson discloses that the registers are formed in a memory (39) coupled to the core processing circuit (31) through a data bus (26).

Regarding claim 3, Anderson discloses that the host processing system comprises a system memory, and wherein the reset vector comprises at least one instruction to fetch data from a system memory (column 2, line 66 through column 3, line 2) coupled to the core processing circuit through the host bridge (17).

Regarding claim 5, Anderson discloses that the host processing system further comprises logic to initiate one or more write bus transactions (lines 7-9 of column 8 mention that the host transfers boot code sequence into the memory. Thus, host initiates the write bus transactions) to load the reset vector in the registers while the core processing circuit is in the reset state (column 6, lines 5-7), and wherein the core processing circuit comprises logic to initiate the system memory in response to execution of the reset vector upon release from the reset state (column 6, lines 13-15). Anderson does not explicitly disclose initiating of one or more write bus transactions at an address translation unit nor one or more read bus transactions at the address translation unit addressed to the system memory, however said transactions would necessarily have to be present in order for the host processing system to properly communicate with the core processing circuit.

Regarding claim 6, Anderson discloses that the host processing system further comprises logic to release the core processing circuit from the reset state in response to loading the reset vector at the boot address (column 2, lines 37-39 and lines 5-11 of column 6).

Regarding claim 9, Anderson discloses a method comprising:

- **having a host processing system** (11) **maintain a core processing unit** (13 comprises core processor 31) **in a reset state** (lines 1-5 of column 6 mentioned that the processor 31 is entered and maintained in reset state by asserting

HRESET through a hardware/software reset. In addition, lines 9-11 of column 3 mention that the releasing of the suspended processor from reset includes host accessing an address. Thus, the core processing system is maintained in a reset state during power up until released by the host) **during power up of the core processing circuit** (lines 4-6 of column 3 mention that the resetting comprises asserting power-on reset signal. In addition, lines 39-43 of column 7 mention that the reset is asserted during power up of the PCMCIA card. Thus, host maintains the core processing circuit in a reset state during power up of the core processing circuit)

- **loading a reset vector to one or more registers at a boot address associated with the core processing circuit** (column 6, lines 13-15 in view of column 8, lines 7-9, Anderson discloses the reset vector as comprising an "initial program," which necessarily must be loaded to registers), **the reset vector comprising one or more instructions to fetch additional instructions from a system memory** (column 8, lines 24-26., Anderson discloses an included loader program to download a program from host 11 to FLASH memory, which necessarily must include fetching additional instructions) **coupled to the core processing circuit through a host bridge** (17) **of a host processing system** (11)
- **setting an address translation unit** ("address decode circuitry" mentioned in line 48 of column 2 is set by host for causing the processor to boot. The booting occurs from the downloaded code within dual port memory as mentioned in lines

5-15 of column 6) **to enable at least one outbound transaction to address at least one location in the system memory to fetch instructions from the system memory in response to requests from the core processing unit** (the host sets the address decode circuitry with a particular address for releasing suspended processor execution and causing the processor to boot (lines 30-33 of column 9). The boot program may include diagnostic routine, which can cause further diagnostic routine to be loaded from host as mentioned in lines 28-35 of column 8. Thus, host sets address decode circuitry or address translation unit to activate booting, which again causes to fetch instructions from the host system memory in response to request from the core processing circuit)

- **in response to an outbound transaction, forwarding the outbound transaction from an internal data bus coupled to the core processing circuit to an external data bus to the system memory** (since additional routine is loaded from the host memory in response to the query from boot program, an outbound transaction is forwarded from internal bus coupled to core processing circuit to external bus coupled to host system memory).

Although Anderson fetches additional code from memory, Anderson does not explicitly disclose an address translation unit addressed to the system memory, however said transactions would necessarily have to be present in order for the host processing system to properly communicate with the core processing circuit.

Futral et al teach a system where ATU (218) is configured to convert an internal data bus address (204) associated with the outbound transaction to an external data bus address (208) and configured to forward the outbound transaction from an internal data bus (lines 47-67 of column 4) coupled to the core processing circuit (202) to an external data bus coupled to the system memory (201).

It would have been obvious for one ordinary skill in the art to incorporate the ATU within Anderson as proper communication between PCMCIA and host requires translating core bus transaction to host bus transaction.

Regarding claim 10, Anderson discloses the method comprising loading the reset vector to a boot address in the registers (column 6, lines 13-15 in view of column 8, lines 7-9; Anderson discloses the reset vector as an "initial program" comprising boot sequences which necessarily must be loaded to registers at a boot location) are formed in a memory (39) coupled to the core processing circuit (31) through a data bus (26).

Regarding claim 12, Anderson discloses the system of claim 3, wherein the host processing system further comprises logic to initiate one or more write bus transactions (lines 7-9 of column 8 mention that the host transfers boot code sequence into the memory. Thus, host initiates the write bus transactions) to load the reset vector in the registers while the core processing circuit is in the reset state (column 6, lines 5-7), and wherein the core processing circuit comprises logic to initiate the system memory in

response to execution of the reset vector upon release from the reset state (column 6, lines 13-15). Anderson does not explicitly disclose initiating of one or more write bus transactions at an address translation unit nor one or more read bus transactions at the address translation unit addressed to the system memory, however said transactions would necessarily have to be present in order for the host processing system to properly communicate with the core processing circuit.

Regarding claim 15, Anderson discloses a method comprising:

- **having a host processing system** (11) **maintain a core processing unit** (13 comprises core processor 31) **in a reset state** (lines 1-5 of column 6 mentioned that the processor 31 is entered and maintained in reset state by asserting HRESET through a hardware/software reset. In addition, lines 9-11 of column 3 mention that the releasing of the suspended processor from reset includes host accessing an address. Thus, the core processing system is maintained in a reset state during power up until released by the host) **during power up of the core processing circuit** (lines 4-6 of column 3 mention that the resetting comprises asserting power-on reset signal. In addition, lines 39-43 of column 7 mention that the reset is asserted during power up of the PCMCIA card. Thus, host maintains the core processing circuit in a reset state during power up of the core processing circuit)
- **loading instructions from a system memory of a host processing system to one or more registers at a boot address associated with the core**

processing circuit (column 6, lines 5-15 in view of column 8, lines 7-9*, Anderson discloses the "initial program" comprising boot code sequences loaded in the core processing circuit, which must necessarily be loaded to registers. In addition, 115 of Fig 5 shows that the host loads the program to the memory of core. Thus, Anderson discloses loading instructions from a system memory of a host processing system to one or more registers at a boot address associated with the core processing circuit)**, the instructions comprising one or more instructions to initialize the core processing circuit upon release of the core processing circuit from the reset state** (lines 5-12 of column 6 mention that the START* by host causes the latch output 61 high, which in turn causes the processor to leave the reset state. These operations are shown in step 117 and 119 in Fig 5. After setting the flip-flop, processor begins booting or initialization as shown in step 121 of Fig 5. Thus, the instructions initialize the core processing circuit upon release of the core processing circuit from the reset state)

- **setting an address translation unit** ("address decode circuitry" mentioned in line 48 of column 2 is set by host for causing the processor to boot. The booting occurs from the downloaded code within dual port memory as mentioned in lines 5-15 of column 6) **to enable at least one outbound transaction to address at least one location in the system memory to fetch instructions from the system memory in response to requests from the core processing unit** (the host sets the address decode circuitry with a particular address for releasing

suspended processor execution and causing the processor to boot (lines 30-33 of column 9). The boot program may include diagnostic routine, which can cause further diagnostic routine to be loaded from host as mentioned in lines 28-35 of column 8. Thus, host sets address decode circuitry or address translation unit to activate booting, which again causes to fetch instructions from the host system memory in response to request from the core processing circuit)

- **in response to an outbound transaction, forwarding the outbound transaction from an internal data bus coupled to the core processing circuit to an external data bus to the system memory** (since additional routine is loaded from the host memory in response to the query from boot program, an outbound transaction is forwarded from internal bus coupled to core processing circuit to external bus coupled to host system memory).

Although Anderson fetches additional code from memory, Anderson does not explicitly disclose an address translation unit addressed to the system memory, however said transactions would necessarily have to be present in order for the host processing system to properly communicate with the core processing circuit.

Futral et al teach a system where ATU (218) is configured to convert an internal data bus address (204) associated with the outbound transaction to an external data bus address (208) and configured to forward the outbound transaction from an internal data bus (lines 47-67 of column 4) coupled to the core processing circuit (202) to an external data bus coupled to the system memory (201).

It would have been obvious for one ordinary skill in the art to incorporate the ATU within Anderson as proper communication between PCMCIA and host requires translating core bus transaction to host bus transaction.

Regarding claim 16, Anderson discloses that the method further comprises transmitting the instructions from the system memory (lines 66-67 of column 2 mention that the PCMCIA downloads code from host) through a host bridge (17) of the host processing system (11).

Regarding claim 17, Anderson discloses the method further comprising releasing the core processing circuit from the reset state in response to loading the instructions at the boot address (column 6, lines 5-12 mention that the processor leaves reset state after the memory is loaded with boot code. In addition, lines 38-39 of column 2 mention that boot logic is configured to release the suspended processor. Thus, the releasing of processor from the reset state is in response to loading the instructions at the boot address).

Regarding claim 18, Anderson discloses the method further comprising loading the instructions to a boot address (lines 10-15 of column 6) in registers formed in a memory (39) coupled to the core processing circuit (31) through a data bus (26).

Regarding claim 22, Anderson discloses **an article comprising:**

- **a storage medium comprising machine-readable instructions encoded there on for** (there must be a storage media to store the instructions necessary to execute the disclosed methods of Anderson)**:**
- **having a host processing system** (11) **maintain a core processing unit** (13) **in a reset state** (lines 1-5 of column 6 mentioned that the processor 31 is entered and maintained in reset state by asserting HRESET through a hardware/software reset. In addition, lines 9-11 of column 3 mention that the releasing of the suspended processor from reset includes host accessing an address. Thus, the core processing system is maintained in a reset state during power up until released by the host) **during power up of the core processing circuit** (lines 4-6 of column 3 mention that the resetting comprises asserting power-on reset signal. In addition, lines 39-43 of column 7 mention that the reset is asserted during power up of the PCMCIA card. Thus, host maintains the core processing circuit in a reset state during power up of the core processing circuit)
- **loading instructions from a system memory of a host processing system to one or more registers at a boot address associated with the core processing circuit** (column 6, lines 5-15 in view of column 8, lines 7-9*, Anderson discloses the "initial program" comprising boot code sequences loaded in the core processing circuit, which must necessarily be loaded to registers. In addition, 115 of Fig 5 shows that the host loads the program to the memory of core. Thus, Anderson discloses loading instructions from a system memory of a host processing system to one or more registers at a boot address associated

with the core processing circuit), **the instructions comprising one or more instructions to initialize the core processing circuit upon release of the core processing circuit from the reset state** (lines 5-12 of column 6 mention that the START* by host causes the latch output 61 high, which in turn causes the processor to leave the reset state. These operations are shown in step 117 and 119 in Fig 5. After setting the flip-flop, processor begins booting or initialization as shown in step 121 of Fig 5. Thus, the instructions initialize the core processing circuit upon release of the core processing circuit from the reset state)

- **setting an address translation unit** ("address decode circuitry" mentioned in line 48 of column 2 is set by host for causing the processor to boot. The booting occurs from the downloaded code within dual port memory as mentioned in lines 5-15 of column 6) **to enable at least one outbound transaction to address at least one location in the system memory to fetch instructions from the system memory in response to requests from the core processing unit** (the host sets the address decode circuitry with a particular address for releasing suspended processor execution and causing the processor to boot (lines 30-33 of column 9). The boot program may include diagnostic routine, which can cause further diagnostic routine to be loaded from host as mentioned in lines 28-35 of column 8. Thus, host sets address decode circuitry or address translation unit to activate booting, which again causes to fetch instructions from the host system memory in response to request from the core processing circuit).

Although Anderson fetches additional code from memory, Anderson does not explicitly disclose an address translation unit addressed to the system memory, however said transactions would necessarily have to be present in order for the host processing system to properly communicate with the core processing circuit.

Futral et al teach a system where ATU (218) is configured to convert an internal data bus address (204) associated with the outbound transaction to an external data bus address (208) and configured to forward the outbound transaction from an internal data bus (lines 47-67 of column 4) coupled to the core processing circuit (202) to an external data bus coupled to the system memory (201).

It would have been obvious for one ordinary skill in the art to incorporate the ATU within Anderson as proper communication between PCMCIA and host requires translating core bus transaction to host bus transaction.

Regarding claim 23, Anderson discloses the method comprising loading the reset vector to a boot address in the registers (column 6, lines 13-15 in view of column 8, lines 7-9; Anderson discloses the reset vector as an "initial program" comprising boot sequences which necessarily must be loaded to registers at a boot location) are formed in a memory (39) coupled to the core processing circuit (31) through a data bus (26).

Regarding claim 25, Anderson discloses the host processing system further comprises logic to initiate one or more write bus transactions (lines 7-9 of column 8 mention that the host transfers boot code sequence into the memory. Thus, host initiates the write bus transactions) to load the reset vector in the registers while the core processing circuit is in the reset state (column 6, lines 5-7). Anderson does not explicitly disclose initiating of one or more write bus transactions at an address translation unit addressed to the system memory, however said transactions would necessarily have to be present in order for the host processing system to properly communicate with the core processing circuit.

Regarding claim 26, Anderson discloses the host processing system further comprises logic to release the core processing circuit from the reset state in response to loading the reset vector at the boot address (column 2, lines 37-39 and lines 5-11 of column 6).

Regarding claim 29, Anderson discloses an article comprising:

- **a storage medium comprising machine-readable instructions encoded there on for** (there must be a storage media to store the instructions necessary to execute the disclosed methods of Anderson):
- **having a host processing system** (11) **maintain a core processing unit** (13) **in a reset state** (lines 1-5 of column 6 mentioned that the processor 31 is entered and maintained in reset state by asserting HRESET through a hardware/software reset. In addition, lines 9-11 of column 3 mention that the

releasing of the suspended processor from reset includes host accessing an address. Thus, the core processing system is maintained in a reset state during power up until released by the host) **during power up of the core processing circuit** (lines 4-6 of column 3 mention that the resetting comprises asserting power-on reset signal. In addition, lines 39-43 of column 7 mention that the reset is asserted during power up of the PCMCIA card. Thus, host maintains the core processing circuit in a reset state during power up of the core processing circuit)

- **loading instructions from a system memory of a host processing system to one or more registers at a boot address associated with the core processing circuit** (column 6, lines 5-15 in view of column 8, lines 7-9*, Anderson discloses the "initial program" comprising boot code sequences loaded in the core processing circuit, which must necessarily be loaded to registers. In addition, 115 of Fig 5 shows that the host loads the program to the memory of core. Thus, Anderson discloses loading instructions from a system memory of a host processing system to one or more registers at a boot address associated with the core processing circuit)**, the instructions comprising one or more instructions to initialize the core processing circuit upon release of the core processing circuit from the reset state** (lines 5-12 of column 6 mention that the START* by host causes the latch output 61 high, which in turn causes the processor to leave the reset state. These operations are shown in step 117 and 119 in Fig 5. After setting the flip-flop, processor begins booting or initialization as shown in step 121 of Fig 5. Thus, the instructions initialize the

core processing circuit upon release of the core processing circuit from the reset state)

- **setting an address translation unit** ("address decode circuitry" mentioned in line 48 of column 2 is set by host for causing the processor to boot. The booting occurs from the downloaded code) **to enable at least one outbound transaction to address at least one location in the system memory to fetch instructions from the system memory in response to requests from the core processing unit** (the host sets the address decode circuitry with a particular address for releasing suspended processor execution and causing the processor to boot (lines 30-33 of column 9). The boot program may include diagnostic routine, which can cause further diagnostic routine to be loaded from host as mentioned in lines 28-35 of column 8. Thus, host sets address decode circuitry or address translation unit to activate booting, which again causes to fetch instructions from the host system memory in response to request from the core processing circuit)

Although Anderson fetches additional code from memory, Anderson does not explicitly disclose an address translation unit addressed to the system memory, however said transactions would necessarily have to be present in order for the host processing system to properly communicate with the core processing circuit.

Futral et al teach a system where ATU (218) is configured to convert an internal data bus address (204) associated with the outbound transaction to an external data bus

address (208) and configured to forward the outbound transaction from an internal data bus (lines 47-67 of column 4) coupled to the core processing circuit (202) to an external data bus coupled to the system memory (201).

It would have been obvious for one ordinary skill in the art to incorporate the ATU within Anderson as proper communication between PCMCIA and host requires translating core bus transaction to host bus transaction.

Regarding claim 30, Anderson discloses that the method further comprises transmitting the instructions from the system memory (lines 66-67 of column 2 mention that the PCMCIA downloads code from host) through a host bridge (17) of the host processing system (11).

Regarding claim 31, Anderson discloses the method further comprising releasing the core processing circuit from the reset state in response to loading the instructions at the boot address (column 6, lines 5-12 mention that the processor leaves reset state after the memory is loaded with boot code. In addition, lines 38-39 of column 2 mention that boot logic is configured to release the suspended processor. Thus, the releasing of processor from the reset state is in response to loading the instructions at the boot address).

Regarding claim 32, Anderson discloses the method further comprising loading the instructions to a boot address (lines 10-15 of column 6) in registers formed in a memory (39) coupled to the core processing circuit (31) through a data bus (26).

Regarding claims 7, 13, 20, 27, and 34, Anderson discloses all of the limitations of respective independent claims 1, 9, 15, 22, and 29, as noted above. However, though Anderson discloses the additional instructions comprising "diagnostic routines" (column 3, lines 13-14), Anderson does not explicitly disclose the system wherein the additional instructions comprise instructions to commence a power-on self test procedure. The examiner takes Official Notice that power-on self test procedures are a well known type of diagnostic routine. It would have been obvious at the time that the invention was made to use power-on self test procedures for the diagnostic routines disclosed by Anderson. The motivation for doing so would have been to assure the integrity of the information stored on the core processing circuit.

4. Claims 8, 14, 21, 28, and 35 rejected under 35 U.S.C. 103(a) as being unpatentable over Anderson (U.S. 5,898,869), in view of Futral et al (US Patent 5925099), further in view of Klein (U.S. 6,216,224).

Regarding claims 8, 14, 21, 28 and 35, Anderson discloses the limitations of respective parent claims 7, 13, 20 27 and 34 as noted above. However, Anderson does not

disclose the method wherein the additional instructions further comprise instructions to launch an operating system to the core processing circuit.

Klein teaches the method wherein the (additional) instructions further comprise instructions to launch an operating system to the core processing circuit (column 4, lines 13-15) in order to "bring the PC up to a state that can be used by a human operator" (page 2, column 4, lines 16-17). It would have been obvious to one of ordinary skill in the ad at the time of invention to incorporate Klein's teachings of loading the operating system with Anderson's disclosure of a system comprising a core processing circuit initialization with a host processing system. The motivation for doing so would have been to accommodate for the use of the system by a human operator.

Response to Arguments

Applicant's arguments filed on 8/8/2006 have been fully considered but they are moot in view of new grounds of rejections.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Fahmida Rahman whose telephone number is 571-272-8159. The examiner can normally be reached on Monday through Friday 8:30 - 5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lynne Browne can be reached on 571-272-3670. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Fahmida Rahman
Examiner
Art Unit 2116

LYNNE H. BROWNE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100